Package, frame rate and color

Русский

Package (remux)

Remux changes the container without re-encoding, where this is possible.

PACKAGE works from the output target. The format of the source is not chosen by hand: the Source is scanned recursively, and the format of each file and its audio codecs are read automatically. Incompatible files are skipped with the reason in the log.

Output video containers:

```
MP4 MOV MKV MXF
```

Output audio containers:

```
M4A ALAC MKA AAC AC3 EAC3 DTS FLAC WAV MP3 OPUS OGG
```

Pipeline:

```
Video Audio Video&Audio
```

All three modes are a pure `copy`: only the stream map (`-map`) and the container change, and the codecs are not touched.

- `Video` keeps only the video tracks (`-map 0:v? -an`).
- `Audio` takes the audio tracks out (`-map 0:a? -vn`) into the selected audio container.
- `Video&Audio` copies video and audio together (`-map 0:v? -map 0:a?`).

A same-container copy is allowed too, if the set of streams changes: for example `MP4 -> MP4 / Video` makes an MP4 without audio, and `MP4 -> MP4 / Video&Audio` makes a clean repackaging

into a suffix folder.

For the audio-only mode the container is checked against the real audio codecs. AAC stereo/mono from MP4/MOV/MKV can go into M4A, AC3/EAC3/DTS into the codec's own file, FLAC/MP3/OPUS into their own containers, ALAC into M4A. MKA stays the universal container for complex or multitrack cases.

For the video and video+audio modes the codec is checked, not only the pair of containers. `MOV -> MP4` is an allowed pair, but it is valid for H.264 and impossible for ProRes: the MP4 muxer has no tag for this codec. Such a file is skipped with a reason like `MP4 does not support copy of prores`, instead of being handed to FFmpeg for a certain error. Practical limits:

- ProRes and DNxHR - in MOV, MXF, MKV; not in MP4;
- HEVC - in MP4, MOV, MKV; not in MXF;
- AV1 and VP9 - in MP4, MKV, WebM; not in MOV or MXF;
- of audio, only PCM 16/24-bit goes into MXF;
- in the `Video&Audio` mode the audio codec is checked the same way.

The tables say what surely works, and a refusal is given only for a measured codec. A codec that is not in the tables is handed to FFmpeg: a blanket ban costs the user a working conversion, and an extra failed run costs one line in the log. `MKV -> MP4` with sound is an ordinary working pair and is not limited in any way; subtitles and data streams do not reach the muxer, because the stream map removes them (`-sn -dn`).

If FFmpeg still breaks off while writing, no empty file stays in the Target: a zero-size result is deleted with a `[CLEAN]` line in the log.

The result is written into the folder of the profile with a recursive mirror of the Source, not into the file name. Example: `output\Remux\remux_mkv_to_mp4_video_audio\Day1\clip.mp4`.

When To Choose Remux

Choose remux if:

- the video/audio streams already fit;
- you need to change the container or the set of streams without an encode;
- you need MP4 faststart;
- you need to repackage MKV/MOV/MXF;
- you must not lose quality on a repeated encode.

Do not choose remux if you need to:

- change the codec;
- change the resolution;
- apply a LUT;
- change the FPS;
- normalize the audio;
- make a downmix.

For this you need the encode/audio/FPS/LUT sections.

Container Compatibility

Not all pairs of codec and container are safe. The window shows only the target containers, and the backend checks each file separately and skips incompatible cases.

In practice:

- MP4 suits H.264/H.265/AAC well.
- MOV is convenient for ProRes/PCM workflows.
- MKV is flexible for archive/storage.
- MXF is needed for broadcast/NLE procedures.
- WEBM with Opus is usually safer to take into OPUS/OGG/MKA for audio-only, or into MKV for video/video+audio.

Frame Rate

The FRAME RATE section does not ask you to type the source FPS by hand. It is read from the video stream.

Fields:

- `Conversion mode`: Varispeed or Conform.
- `Target FPS`: **16, 18, 23.976, 24, 25, 29.97, 30, 48, 50, 59.94, 60**. Fractional rates are written as exact fractions (`24000/1001`, `30000/1001`, `60000/1001`); a rate that could not be read is rejected, not replaced with a default. 120/144/240 are left out on purpose: this is not a gaming tool. 16 and 18 are for silent film and amateur 8 mm.

- **The source rate can be anything**, up to high-speed footage: 8000 fps → 24 went through without losing a single frame (2000 frames, exactly 83.333 s).
- `Output profile`: H.264 MP4, H.265 MP4, ProRes MOV/MXF, DNxHR MOV/MXF.
- For H.264/H.265 the AAC bitrate is available.
- For ProRes/DNxHR, PCM 16/24/32-float in Apple MOV is available; MOV keeps the selected working rate up to 176.4/192 kHz.
- MXF takes PCM 16/24, but the muxer of the FFmpeg build in use finalizes audio only at 48 kHz; PCM 32-float in MXF is turned off.
- Audio oversampling: COPY, 2x 88.2/96 or 4x 176.4/192 through SoX/libsoxr. For MXF this is an internal working rate with a mandatory final resampling to 48 kHz.
- The CLI FPS wizard offers the same six profiles, bit depths and `x1/x2/x4`; the console menu and the window build the command with the shared `system_core\core\fps_contract.py`.

Varispeed

Varispeed changes the playback speed. The duration changes; the motion stays “clean”, without frame interpolation.

Use it when:

- this is a deliberate conform by speed;
- the audio must stretch/shrink too;
- you need to convert the material timing.

Conform

Conform changes the interpretation/output rate, and its meaning is not the same as that of varispeed. Choose it when the workflow needs a conform output exactly, not a speed change.

Before a batch operation, check one file and look closely at the duration.

Color and LUT

The COLOR section applies LUTs and grading presets.

Main elements:

- the list of LUTs as rows;

- the buttons of the color profiles, which are the only choice of the mode;
- `Pre-LUT gamma`;
- output targets;
- encode/decode backend;
- audio/container/resolution/pixel format/CRF/CQ.

LUT Cache

The LUT is chosen from `LUTs\` as a list of rows: the pinned ones first, then the selected one, the recent ones and the rest. The window can:

- choose a LUT;
- pin a frequently used LUT (the pin at the row);
- add LUT files (`.cube`, `.3dl`, `.lut`); they are copied into `LUTs`;
- add a folder with LUTs; it is copied into `LUTs` with its subfolders;
- open the `LUTs` folder.

A file with the same name in `LUTs` is replaced silently. If no LUT is selected, `LUTs\active.cube` is taken.

Color Operations

Available:

- `LUT only`;
- `Pre-expose + LUT`;
- `Pregamma + LUT`;
- `HDR to SDR Hable`.

`LUT only` applies the LUT without the pre-gamma helper.

`Pre-expose + LUT` and `Pregamma + LUT` technically use one parameter, `eq=gamma=<number>`, before the LUT, but with different starting values and a different creative meaning.

`HDR to SDR Hable` applies a tonemap-style transform for SDR output.

The service field `script_preset` is kept hidden and is passed to the backend by the selected profile button. There is no separate duplicate choice `Color operation` in the window.

18% Gray Preview

For pregamma/pre-expose the window shows a preview of 18% gray:

- on the left, the source gray;
- on the right, the result before the LUT;
- the caption says whether the gray becomes lighter, darker or stays neutral.

This does not replace viewing the final video, but it is a good quick indicator of the direction of the correction.

LUT Path Escaping

Windows paths to a LUT can contain:

- the drive `C:`;
- spaces;
- apostrophes;
- commas;
- semicolons;
- brackets.

The window and the wrappers use one shared helper that escapes the FFmpeg filter graph. If a LUT works through the window, it works through the matching wrapper too.

Output Targets

Color targets:

- x264 / H.264;
- x265 / HEVC;
- SVT-AV1;
- ProRes 422;
- ProRes LT.

To check the color, it is better to start with a short H.264 preview, not straight away with a heavy ProRes/AV1 batch.

Recommended LUT Procedure

1. Choose the Source and the Target.
2. Choose the LUT.
3. Choose the profile.
4. Check the 18% gray preview, if it is available.
5. Press COMMAND.
6. Turn on `Test on first file`.
7. Judge the short result.
8. Run the batch.

Typical Errors

LUT Was Not Applied

Check:

- whether a `.cube` is selected;
- whether the file exists;
- whether an old/wrong pinned LUT is in the way;
- that the dry run contains the LUT filter.

Color Too Dark Or Too Light

Check `Pre-LUT gamma`. For different LUTs the starting value may need a manual change.

FPS Result Has The Wrong Duration

Compare Varispeed and Conform. These are not cosmetic modes: they affect the timing differently.